1

LEVEL II

ADA084289

UNIVERSITY OF MARYLAND

COMPUTER SCIENCE CENTER

COLLEGE PARK, MARYLAND
20742

DDC FILE COPY

DTIC
ELECTE
MAY 19 1980
S D

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

80 5 19 119

(9) Technical rept.,

(15) DAAG53-76-C-∅138,
DARPA Order-32∅6

(14) TR-769
DAAG-53-76C-0138

(11) May [redacted]79

Accession For	
NTIS GRA&I	☑
DDC TAB	□
Unannounced	□
Justification______	
By______	
Distribution/	
Availability Codes	
Dist A	Avail and/or special

(6) COMPUTING THE EULER NUMBER OF AN IMAGE FROM ITS QUADTREE

(10) Charles R. Dyer
Computer Science Center
University of Maryland
College Park, MD 20742

(12) 17

ABSTRACT

An algorithm is presented which computes the Euler number, i.e., the number of components minus the number of holes, of a binary image represented by a quadtree. The local property counting techniques used with an array representation are generalized to counting local node configurations in a quadtree. The average worst case running time of the algorithm is proportional to the product of the number of nodes representing the components and the logarithm of the image diameter.

The support of the Defense Advanced Research Projects Agency and the U.S. Army Night Vision Laboratory under Contract DAAG-53-76C-0138 (DARPA Order 3206) is gratefully acknowledged, as is the help of Kathryn Riley in preparing this paper. The helpful comments of Hanan Samet and Azriel Rosenfeld are also greatly appreciated.

403018 JOB

1. Introduction

In this paper we continue our investigation of the quadtree representation as a suitable data structure for efficiently performing various operations on images. Earlier results showed that converting between quadtree and chain code representations [1,2], measuring the perimeter of a region stored in a quadtree [3], and connected component labeling in a quadtree [4] can all be accomplished efficiently.

We now consider the problem of computing the Euler number (or genus), i.e., the number of components minus the number of holes, of a binary image which is stored in a quadtree. This topological property is well known to be locally countable when the image is stored using either a rectangular [5,6] or hexagonal [7] array representation. For example, assume that a binary image is stored in a rectangular array. Let the 1's be regarded as connected to their four horizontal and vertical neighbors, while the 0's are regarded as connected to all eight of their neighbors. Define V to be the number of 1's in the image, E the number of horizontally or vertically adjacent pairs of 1's, and F the number of 2 by 2 blocks of 1's. Then it can be proved by induction on V that the Euler number of the image is equal to V-E+F.

The quadtree of a 2^n by 2^n binary image is defined recursively as follows. Let the root of the quadtree be associated with the entire image; the level of the root is n. If the 2^k by 2^k block of the image associated with an arbitrary node at level k does not consist of either all 1's or all 0's, then subdivide the block into four 2^{k-1} by 2^{k-1} quadrants and associate these subblocks with four nodes designated as the four sons of the given node; each son is considered to be at level k-1.

Each node in a quadtree is stored as a record containing six fields. The first five fields contain pointers to a node's father and four sons. The sixth field describes the contents of the subimage associated with the node--WHITE if the image is all 0's, BLACK if it contains all 1's, and GRAY otherwise. Readily, all non-terminal nodes are GRAY and all terminal nodes are either BLACK or WHITE.

While it is possible to compute the Euler number by modifying the connected component labeling algorithm [4] to simultaneously label components of 1's or 0's, the algorithm presented here is simpler and faster, making use of a generalization of the V-E+F formula to "pixels" of arbitrary size.

2. Euler number from a quadtree

The formula quoted in Section 1 for computing the Euler number considered the image to be divided into unit-sized pixels. In a quadtree representation, the "pixel" size is variable, i.e., each leaf node corresponds to a 2^k by 2^k block of the image with constant brightness (either 0 or 1 in our case where the image is binary). After defining some notation, we show that the V-E+F formula for rectangular arrays with unit-sized pixels generalizes to quadtrees where an image consists of pixels having variable sizes and positions (powers of 2).

Two nodes are said to be adjacent if their blocks of the image share a common side as shown in Figure 1a-b. Two nodes which touch at a corner only (Figure 1c) are not considered adjacent. Notice that by definition of a quadtree, adjacent nodes cannot properly overlap. A group of nodes are said to surround a point if there exists a 2 by 2 block of pixels such that each node's block contains at least one of the four pixels, and the union of the nodes' blocks contains all four of them. The four possible ways that one, two, three or four nodes can surround a point are shown in Figure 2.

We now state and prove our main result:

Theorem 1. Given a quadtree with B BLACK nodes, A pairs of adjacent BLACK nodes, and S triples or quadruples of BLACK nodes which surround a point, the Euler number of the binary image which it represents is equal to B-A+S.

Proof: By induction on the number of BLACK nodes B. We show that for every possible way of adding a new BLACK node to a quadtree, the relation B-A+S = V-E+F remains true, where the Euler number V-E+F is computed on the original binary image from which the given quadtree was constructed.

A quadtree containing a single BLACK node represents a 2^k by 2^k connected block of 1's in the image. In this case B-A+S = 1-0+0 = 1, and the Euler number for this image is $V-E+F = 2^{2k}-(2^{k+1}(2^k-1)) + (2^k-1)^2 = 1$. Adding a BLACK node which is not adjacent to any BLACK node already in the quadtree similarly increases V-E+F by 1 since the new block of 1's is a new component; B-A+S is also increased by 1 since B increases by 1 and A and S are unchanged.

Adding a BLACK node of size 2^k by 2^k which is adjacent to exactly one other BLACK node, say of size 2^j by 2^j, implies that V-E+F remains unchanged since V increases by 2^{2k}; E increases by $2^{\min(j,k)}$, the length of the common side, plus $(2^{k+1}(2^k-1))$, the number of edges within the 2^k by 2^k block; and F increases by the number of 2 by 2 blocks inside the new block, i.e., $(2^k-1)^2$, plus $2^{\min(j,k)}-1$, the number of 2 by 2 blocks which span the boundary between the two node's blocks. Summing, we see that V-E+F increases by $2^{2k}-(2^{\min(j,k)}+2^{k+1}(2^k-1)) + ((2^k-1)^2 + 2^{\min(j,k)}-1) = 0$. Similarly, B-A+S remains unchanged since B and A increase by 1 and S does not change.

In general, if the new BLACK node is adjacent along one of its sides to m other BLACK nodes, no two of which are adjacent (see Figure 3), then V increases by 2^{2k}, E increases by $(2^{k+1}(2^k-1))$ plus the sum of the lengths of the adjacent sides, and F increases by $(2^k-1)^2$ plus the number of new 2 by 2 BLACK blocks overlapping adjacent block boundaries, i.e., the sum of the lengths of the adjacent sides minus m. Summing, V-E+F increases by 1-m. On the other hand, B increases by 1, A increases by m, and S is unchanged, so B-A+S also increases by 1-m.

Now consider adding a BLACK node of size 2^k by 2^k which is adjacent along one of its sides to a pair of BLACK nodes, of sizes 2^i by 2^i and 2^j by 2^j, which touch along a common side (see Figure 4). In this case V-E+F is unchanged since V increases by 2^{2k}, E increases by $(2^{k+1}(2^k-1))+2^i+2^j$, and F increases by $(2^k-1)^2+2^i+2^j-1$. Likewise, B-A+S is unchanged since B increases by 1, A increases by 2 and S increases by 1. In general, given m BLACK nodes $n_1,n_2,\dots,n_m$ such that n_i is adjacent to n_{i-1} for $1<i\leq m$, adding a new BLACK node which is adjacent to all of them (Figure 5) implies that B-A+S is again unchanged since B increases by 1, A increases by m and S increases by m-1. On the other hand, it is easily shown that V-E+F also remains unchanged.

Next, consider the case where the new BLACK node fills in a corner, i.e., a point becomes surrounded by BLACK nodes.

There are two ways that this can happen, with either three or four BLACK nodes surrounding the point as shown in Figure 2c-d. In the situation where four BLACK nodes surround a point, assume that the new node's block is 2^k by 2^k and the other three nodes' blocks are 2^h by 2^h, 2^i by 2^i, and 2^j by 2^j.* Then V increases by 2^{2k}, E increases by $(2^{k+1}(2^k-1)) + 2^{\min(h,k)} + 2^{\min(j,k)}$, and F increases by $(2^k-1)^2 + (2^{\min(h,k)}-1) + (2^{\min(j,k)}-1) + 1$. Summing, we see that V-E+F is unchanged by the addition of this new block of 1's in the image.
A similar argument shows that V-E+F is also unchanged when three BLACK nodes surround a point. Likewise, in either situation B-A+S remains unchanged by the addition of this BLACK node in the quadtree since B increases by 1, A increases by 2, and S increases by 1.

Note that combinations of these cases on one or more sides of a node are linear sums of the contributions of each type of node adjacency and hence the induction argument also holds for these variations.

Q.E.D.

* See Figure 6.

3. The algorithm

Given the technique described in Section 2 for computing the Euler number, this section informally describes an algorithm which traverses a quadtree and accumulates the value of B-A+S. The algorithm is analogous to phase one of the connected component labeling algorithm given in [4] in that the procedure is built around finding adjacent pairs of BLACK nodes. The reader is referred to [4] for details which are not included in the sketch given here.

Given a quadtree derived from a 2^n by 2^n image, the algorithm traverses the tree in postorder, visiting sons in the order NW, NE, SW, and SE. At each BLACK node, B is increased by 1 and all of the leaf nodes which are adjacent to the node's eastern and southern sides are checked. For each BLACK neighbor, A is incremented by 1. S is incremented for every successive pair of BLACK neighbors along a side since this is an instance of the configuration shown in Figure 4. Finally, the nodes surrounding the southeast corner point of the given node are checked; if they are all BLACK, S is increased by 1 since this is an instance of the configurations shown in Figures 2c-d.

This procedure guarantees that each BLACK node, each pair of adjacent BLACK nodes, and each triple or quadruple of BLACK nodes surrounding a point will be discovered and counted exactly once. To see this, note that due to the traversal order, by the time a BLACK node is visited, its northern and western BLACK node adjacencies have already been checked. Thus the additions to A and S resulting from pairs of BLACK nodes along the given node's northern and western sides, and triples or quadruples of BLACK nodes surrounding the given node's northwest, northeast, and southwest corners, have previously been counted.

Phase one of the connected component labeling algorithm [4] can be modified to compute the Euler number of the image without affecting the asymptotic running time of the algorithm. While the computation of B and A are readily included in that algorithm, extensions must be made in order that the value of S can be computed. Additions to S resulting from successive neighboring BLACK nodes can be included by retaining the color of the previously checked neighbor during the sequential scans of eastern and southern adjacencies. The only other way that S can be increased is if the southeast corner point is surrounded by BLACK nodes. If the easternmost of the given node's southern neighbors and the southermost of its eastern neighbors are BLACK, it is necessary to check at most one more leaf node (depending on whether or not either of these neighbors extends past the corner), which is located diagonally across from the southeast corner, in order to test this case.

The asymptotic running time of the connected component labeling algorithm is clearly not affected by the extensions sketched here for computing B, A and S. Thus from [4], we can immediately conclude that the worst case average execution time for computing the Euler number from a quadtree is proportional to the product of the number of BLACK nodes and the logarithm of the image diameter.

4. Concluding remarks

An algorithm has been presented for computing the Euler number of a binary image represented by a quadtree using a generalization of the local counting technique used with an array representation. The algorithm's running time is proportional to Bn, where B is the number of BLACK nodes and the image size is 2^n by 2^n. For many images, this compares very favorably with the $O(4^n)$ time algorithm which uses the array representation.

The algorithm given here considers points in a component to be connected to their four horizontal and vertical neighbors. Alternatively, if we assume each 1 to be connected to its eight neighbors, an analogous Euler formula exists which takes into account occurrences of the patterns $\begin{array}{cc}1 & \\ & 1\end{array}$ and $\begin{array}{cc}1 & 1 \\ & 1\end{array}$, and all 90° rotations of each, as well as the previous patterns [8]. Similarly, in quadtrees we can additionally count pairs of diagonally adjacent BLACK nodes (Figure 7) and pairs or triples of BLACK nodes forming concave corners (Figure 8). The asymptotic running time of the algorithm will not be affected by this modification.

References

1. C. R. Dyer, A. Rosenfeld, and H. Samet, Region representation: boundary codes from quadtrees, Computer Science Technical Report TR-732, University of Maryland, College Park, MD, February 1979.

2. H. Samet, Region representation: quadtrees from boundary codes, Computer Science Technical Report TR-741, University of Maryland, College Park, MD, March 1979.

3. H. Samet, Computing perimeters of images represented by quadtrees, Computer Science Technical Report TR-755, University of Maryland, College Park, MD, April 1979.

4. H. Samet, Connected component labeling using quadtrees, Computer Science Technical Report TR-756, University of Maryland, College Park, MD, April 1979.

5. A. Rosenfeld, *Picture Processing by Computer*, Academic Press, New York, 1969.

6. M. Minsky and S. Papert, *Perceptrons*, The MIT Press, Cambridge, MA, 1969.

7. S. B. Gray, Local properties of binary images in two dimensions, *IEEE Trans. Computers 20*, 1971, 551-561.

8. A. Rosenfeld, *Picture Languages*, Academic Press, New York, 1979.

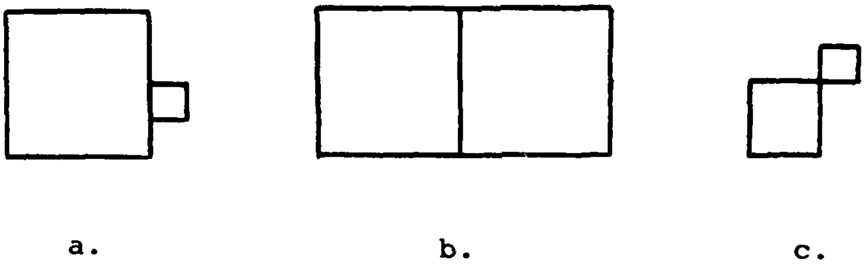

Figure 1. Pairs of adjacent (a-b) and non-adjacent (c) nodes.

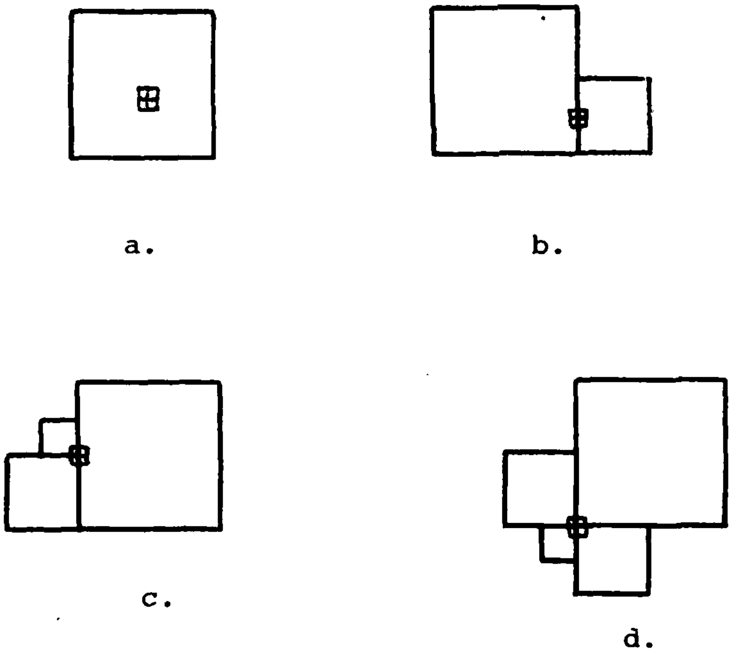

Figure 2. Possible configurations of up to four nodes which surround a point (the midpoints of the small 2 by 2 blocks).

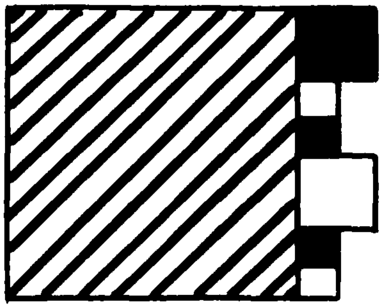

Figure 3. New BLACK node (striped) which is adjacent along its east side to three other BLACK nodes (solid black), no two of which are adjacent.

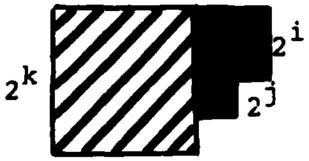

Figure 4. New BLACK node (striped) which is adjacent along its east side to two BLACK nodes (solid black). The two old nodes are adjacent.

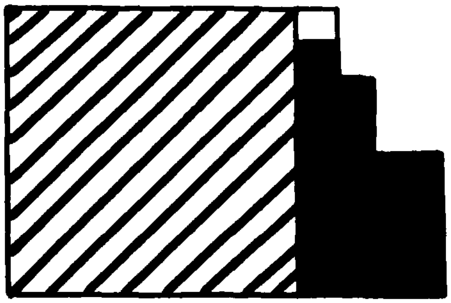

Figure 5. New BLACK node (striped) which is adjacent along its east side to three consecutively adjacent BLACK nodes (solid black).

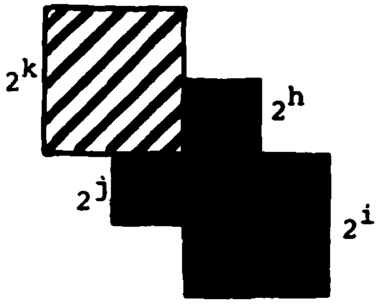

Figure 6. New BLACK node's southeast corner is surrounded by four BLACK nodes. The new block is striped, the old blocks are solid.

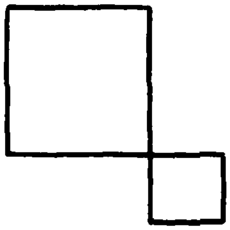

Figure 7. A diagonally adjacent pair of nodes.

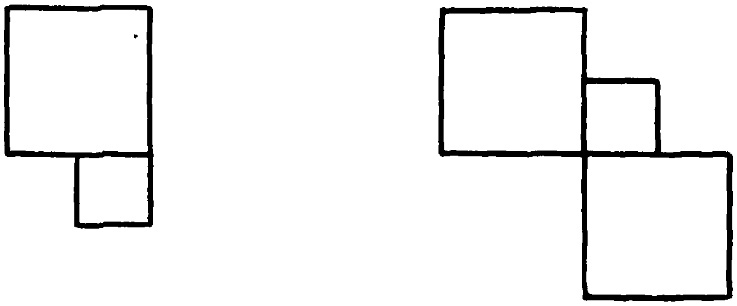

Figure 8. A pair and triple of nodes forming concave corners.

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 2. GOVT ACCESSION NO. A084289	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) COMPUTING THE EULER NUMBER OF AN IMAGE FROM ITS QUADTREE	5. TYPE OF REPORT & PERIOD COVERED Technical
	6. PERFORMING ORG. REPORT NUMBER TR-769
7. AUTHOR(s) Charles R. Dyer	8. CONTRACT OR GRANT NUMBER(s) DAAG-53-76C-0138
9. PERFORMING ORGANIZATION NAME AND ADDRESS Computer Science Center University of Maryland College Park, MD 20742	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS U.S. Army Night Vision Laboratory Fort Belvoir, VA 22060	12. REPORT DATE May 1979
	13. NUMBER OF PAGES 16
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report) Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Pattern recognition Euler number
Image processing Genus
Topology Quadtrees

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

An algorithm is presented which computes the Euler number, i.e., the number of components minus the number of holes, of a binary image represented by a quadtree. The local property counting techniques used with an array representation are generalized to counting local node configurations in a quadtree. The average worst case running time of the algorithm is proportional to the product of the number of nodes representing the components and the logarithm of the image diameter.

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)